Create account Log in

Article Talk Read Edit View history Search

WIKIPEDIA
The Free Encyclopedia

Navigation
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Data item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Čeština
Deutsch
Français
Latviešu
Bahasa Melayu
Polski
Русский
Српски / srpski
Suomi
中文
Edit links

Internet media type

From Wikipedia, the free encyclopedia

An **Internet media type**[1] is a standard identifier used on the Internet to indicate the type of data that a file contains. Common uses include the following:

- email clients use them to identify attachment files,
- web browsers use them to determine how to display or output files that are not in HTML format,
- search engines use them to classify data files on the web.

A media type is composed of a *type*, a *subtype*, and zero or more optional parameters. As an example, an HTML file might be designated `text/html; charset=UTF-8`. In this example `text` is the type, `html` is the subtype, and `charset=UTF-8` is an optional parameter indicating the character encoding.

IANA manages the official registry of media types .

The identifiers were originally defined in RFC 2046 , and were called **MIME types** because they referred to the non-ASCII parts of email messages that were composed using the MIME (Multipurpose Internet Mail Extensions) specification. They are also sometimes referred to as **Content-types**.

Their use has expanded from email sent through SMTP, to other protocols such as HTTP, RTP and SIP.

New media types can be created with the procedures outlined in RFC 6838 .

Contents

1 Limitations
2 Naming
2.1 Registration trees
2.1.1 Standards tree
2.1.2 Vendor tree
2.1.3 Personal or Vanity tree
2.1.4 Unregistered x. tree
2.2 Suffix
3 List of common media types
3.1 Type application
3.2 Type audio
3.3 Type example
3.4 Type image
3.5 Type message
3.6 Type model
3.7 Type multipart
3.8 Type text
3.9 Type video
4 List of common media subtype prefixes
4.1 Prefix vnd
4.2 Prefix x
5 See also
6 References
7 External links

Limitations [edit]

Internet media types are often used as part of a communication protocol between two applications (the *source* and *destination*). In this context, internet media type specifiers experience several problems.

The first problem is the ability of the source application (i.e. web server, email client) to correctly determine an internet media type for a piece of content. Many applications attempt to heuristically classify a file using its filename extension or with magic numbers. Neither approach is perfect, and may incorrectly classify a content's media type:

- Incorrect filename extension: a filename extension classifier will report an incorrect media type. For instance, some applications incorrectly give Rich text format files the .doc file extensions, instead of the correct .rtf extension.
- No filename extension: a filename extension classifier will report no media type, or will (incorrectly) report a catch-

all type such as `application/octet-stream`. Files without extension are common on unix systems.

- Filename extension collisions: when multiple formats use the same filename extension, a filename extension classifier will choose one media type arbitrarily. For instance, both Microsoft Word templates and graphviz graph files use the extension .dot.
- Ambiguous container formats: a magic number classifier may give a correct, though non-specific, media type, thus preventing a meaningful interpretation of the content. For instance, Office Open XML (.docx) format and Java executable (.jar) are both implemented internally as a zipped archive. A magic number system may classify such files as `application/zip` instead of the more specific type. Similar problems occur between XML and application formats implemented on top of XML.
- Ambiguous magic numbers: an attacker can create a file which is identified simultaneously as two separate internet media types. For instance, the internal structure of a Gifar makes it both a valid GIF image and Java executable.

The second problem is the destination application's ability to trust the internet media type reported by the sender. As above, the internet media type is incorrect in some circumstances, and must be treated with skepticism. As early as 2002, the W3C unambiguously warned that it is a "serious error" if internet media type is incorrect, and that software should not attempt to guess a correct media type.[1]:Section 2 Nonetheless, software engineering principles encourage software that forgives a certain degree of malformed input, and user experience suffers when software fails to correctly interpret the content. Consequently, the many destination applications are designed to attempt recovery from such errors and identify a correct media type.[2][3]

The destination application has no more knowledge of the content than the source application, and attempts to infer the media type at the destination are equally difficult. This can lead to incompatibilities between source and destination applications, and in the worst-case, security vulnerabilities such as the Gifar attack or Cross-site scripting attacks.[4][5] Advanced content sniffing approaches have been proposed to balance interoperability and security in such situations.[3]

Naming [edit]

Media type consists of top-level type name and subtype name, which is further structured into so-called "trees". Media types can optionally define companion data, known as parameters.

```
top-level type name / subtype name [ ; parameters ]
```

```
top-level type name / [ tree. ] subtype name [ +suffix ] [ ; parameters ]
```

The currently registered top-level type names are: application, audio, example, image, message, model, multipart, text, video.

Subtype name typically consists of a media type name, but it may or must also contain other content, such as tree prefix (facet), producer's name, product name or suffix - according the different rules in registration trees.

Examples: "image/svg+xml", "application/vnd.oasis.opendocument.text", "text/plain; charset=utf-8", "video/mp4; codecs="avc1.640028""[6]

Registration trees [edit]

All media types should be registered using the IANA registration procedures. For the efficiency and flexibility of the media type registration process, different structures of subtype names can be registered in registration "trees" that are distinguished by the use of faceted names, e.g. subtype names that begin with a "tree." prefix (facet). Currently the following trees are created: standard, vendor, personal or vanity, unregistered "x.". These registration trees were first defined in November 1996 (obsoleted RFC 2048 - currently RFC 6838). New registration trees may be created by IETF Standards Action - for external registration and management by well-known permanent organizations (e.g. scientific societies).

Standards tree [edit]

Media types in the standards tree do not use any tree facet (prefix).

Registrations in the standards tree must be either associated with IETF specifications approved directly by the IESG, or registered by a IANA-recognized standards-related organization.

```
type / media type name [+suffix]
```

Examples: "application/xhtml+xml", "image/png"

Vendor tree [edit]

Vendor tree is used for media types associated with publicly available products. It uses "vnd." facet.

The terms "vendor" and "producer" are considered equivalent in the context. Industry consortia as well as non-commercial entities can register media types in the vendor tree. A registration in the vendor tree may be created by anyone who needs to interchange files associated with some software product or set of products. However, the

registration belongs to the vendor or organization producing the software that employs the type being registered, and that vendor or organization can at any time elect to assert ownership of a registration done by a third party.

`type / vnd. media type name [+suffix]` - used in the case of well-known producer

`type / vnd. producer's name followed by media type name [+suffix]` - producer's name must be approved by IANA

`type / vnd. producer's name followed by product's name [+suffix]` - producer's name must be approved by IANA

Examples: "application/vnd.ms-excel", "application/vnd.oasis.opendocument.text" for OASIS OpenDocument Text, "application/vnd.etsi.asic-s+zip" for ETSI ASiC

Personal or Vanity tree [edit]

Personal or Vanity tree includes media types created experimentally or as part of products that are not distributed commercially. It uses "prs." facet.

`type / prs. media type name [+suffix]`

Unregistered x. tree [edit]

The "x." tree may be used for media types intended exclusively for use in private, local environments and only with the active agreement of the parties exchanging them. Types in this tree cannot be registered.

According to RFC 6838 (published in January 2013), any use of types in the "x." tree is strongly discouraged. Media types with names beginning with "x-" are no longer considered to be members of this tree since January 2013.

According to the previous version of RFC 6838 - obsoleted RFC 2048 (published in November 1996) it should rarely, if ever, be necessary to use unregistered experimental types, and as such use of both "x-" and "x." forms is discouraged. Previous versions of that RFC - RFC 1590 and RFC 1521 stated that the use of "x-" notation for the subtype name may be used for unregistered and private subtypes, but this recommendation was obsoleted in November 1996.

All media types should be registered using the simplified IANA registration procedures for vendor and personal trees or using the standards procedure for standards tree.

Media types that have been widely deployed (with an unfaceted subtype name beginning with the "x-" prefix) without being registered, should be, if possible, reregistered with a proper faceted subtype name. If this is not possible, the media type can, after an approval by both the media types reviewer and the IESG, be registered in the proper tree with its unfaceted name.

`type / x. media type name [+suffix]`

Suffix [edit]

Suffix is an augmentation to the media type definition to additionally specify the underlying structure of that media type. Media types that make use of a named structured syntax should use the appropriate IANA-registered "+suffix" for that structured syntax when they are registered. Unregistered suffixes should not be used (since January 2013). Structured syntax suffix registration procedures are defined in RFC 6838 .

The currently registered suffixes are: (in RFC 6839) +xml, +json, +ber, +der, +fastinfoset, +wbxml, +zip, (in RFC 7049) +cbor[7]

"+xml" suffix is defined since January 2001 (RFC 3023). Formal registration of "+xml" suffix and other suffixes is defined since January 2013 (RFC 6839).

List of common media types [edit]

IANA manages the official registry of media types . Among others, it includes the following types:

Type `application` [edit]

For Multipurpose files:

- `application/atom+xml`: Atom feeds
- `application/ecmascript`: ECMAScript/JavaScript; Defined in RFC 4329 (equivalent to `application/javascript` but with stricter processing rules)
- `application/EDI-X12`: EDI X12 data; Defined in RFC 1767
- `application/EDIFACT`: EDI EDIFACT data; Defined in RFC 1767
- `application/json`: JavaScript Object Notation JSON; Defined in RFC 4627
- `application/javascript`: ECMAScript/JavaScript; Defined in RFC 4329 (equivalent to `application/ecmascript` but with looser processing rules) It is not accepted in IE 8 or earlier -

text/javascript is accepted but it is defined as obsolete in RFC 4329 . The "type" attribute of the <script> tag in HTML5 is optional. In practice, omitting the media type of JavaScript programs is the most interoperable solution, since all browsers have always assumed the correct default even before HTML5.

- application/octet-stream: Arbitrary binary data.[8] Generally speaking this type identifies files that are not associated with a specific application. Contrary to past assumptions by software packages such as Apache this is not a type that should be applied to unknown files. In such a case, a server or application should not indicate a content type, as it may be incorrect, but rather, should omit the type in order to allow the recipient to guess the type.[9]
- application/ogg: Ogg, a multimedia bitstream container format; Defined in RFC 5334
- application/pdf: Portable Document Format, PDF has been in use for document exchange on the Internet since 1993; Defined in RFC 3778
- application/postscript: PostScript; Defined in RFC 2046
- application/rdf+xml: Resource Description Framework; Defined by RFC 3870
- application/rss+xml: RSS feeds
- application/soap+xml: SOAP; Defined by RFC 3902
- application/font-woff: Web Open Font Format; (candidate recommendation; use application/x-font-woff until standard is official)
- application/xhtml+xml: XHTML; Defined by RFC 3236
- application/xml: XML files; Defined by RFC 3023
- application/xml-dtd: DTD files; Defined by RFC 3023
- application/xop+xml: XOP
- application/zip: ZIP archive files; Registered[10]
- application/gzip: Gzip, Defined in RFC 6713
- application/example: example in documentation, Defined in RFC 4735
- application/x-nacl: for Native Client modules the type must be “application/x-nacl”

Type audio [edit]

For Audio.

- audio/basic: μ-law audio at 8 kHz, 1 channel; Defined in RFC 2046
- audio/L24: 24bit Linear PCM audio at 8–48 kHz, 1-N channels; Defined in RFC 3190
- audio/mp4: MP4 audio
- audio/mpeg: MP3 or other MPEG audio; Defined in RFC 3003
- audio/ogg: Ogg Vorbis, Speex, Flac and other audio; Defined in RFC 5334
- audio/opus: Opus audio
- audio/vorbis: Vorbis encoded audio; Defined in RFC 5215
- audio/vnd.rn-realaudio: RealAudio; Documented in RealPlayer Help[11]
- audio/vnd.wave: WAV audio; Defined in RFC 2361
- audio/webm: WebM open media format
- audio/example: example in documentation, Defined in RFC 4735

Type example [edit]

For example types in documentation, not for real code.

- example: Defined in RFC 4735

Type image [edit]

- image/gif: GIF image; Defined in RFC 2045 and RFC 2046
- image/jpeg: JPEG JFIF image; Defined in RFC 2045 and RFC 2046
- image/pjpeg: JPEG JFIF image; Associated with Internet Explorer; Listed in ms775147(v=vs.85) - Progressive JPEG, initiated before global browser support for progressive JPEGs (Microsoft and Firefox).
- image/png: Portable Network Graphics; Registered,[12] Defined in RFC 2083
- image/svg+xml: SVG vector image; Defined in SVG Tiny 1.2 Specification Appendix M
- image/vnd.djvu : DjVu image and multipage document format. [13]
- image/example: example in documentation, Defined in RFC 4735

Type message [edit]

- message/http: Defined in RFC 2616
- message/imdn+xml: IMDN Instant Message Disposition Notification; Defined in RFC 5438
- message/partial: Email; Defined in RFC 2045 and RFC 2046
- message/rfc822: Email; EML files, MIME files, MHT files, MHTML files; Defined in RFC 2045 and RFC 2046
- message/example: example in documentation, Defined in RFC 4735

Type `model` [edit]

For 3D models.

- `model/iges`: IGS files, IGES files; Defined in RFC 2077
- `model/mesh`: MSH files, MESH files; Defined in RFC 2077 , SILO files
- `model/vrml`: WRL files, VRML files; Defined in RFC 2077
- `model/x3d+binary`: X3D ISO standard for representing 3D computer graphics, X3DB binary files - never Internet Assigned Numbers Authority approved
- `model/x3d+fastinfoset`: X3D ISO standard for representing 3D computer graphics, X3DB binary files (application in process, this replaces any use of `model/x3d+binary`)
- `model/x3d-vrml`: X3D ISO standard for representing 3D computer graphics, X3DV VRML files (application in process, previous uses may have been `model/x3d+vrml`)
- `model/x3d+xml`: X3D ISO standard for representing 3D computer graphics, X3D XML files
- `model/example`: example in documentation, Defined in RFC 4735

Type `multipart` [edit]

For archives and other objects made of more than one part.

- `multipart/mixed`: MIME Email; Defined in RFC 2045 and RFC 2046
- `multipart/alternative`: MIME Email; Defined in RFC 2045 and RFC 2046
- `multipart/related`: MIME Email; Defined in RFC 2387 and used by MHTML (HTML mail)
- `multipart/form-data`: MIME Webform; Defined in RFC 2388
- `multipart/signed`: Defined in RFC 1847
- `multipart/encrypted`: Defined in RFC 1847
- `multipart/example`: example in documentation, Defined in RFC 4735

Type `text` [edit]

For human-readable text and source code.

- `text/cmd`: commands; subtype resident in Gecko browsers like Firefox 3.5
- `text/css`: Cascading Style Sheets; Defined in RFC 2318
- `text/csv`: Comma-separated values; Defined in RFC 4180
- `text/example`: example in documentation, Defined in RFC 4735
- `text/html`: HTML; Defined in RFC 2854
- `text/javascript` (**Obsolete**): JavaScript; Defined in and made obsolete in RFC 4329 in order to discourage its usage in favor of `application/javascript`. However, `text/javascript` is allowed in HTML 4 and 5 and, unlike `application/javascript`, has cross-browser support. The "type" attribute of the `<script>` tag in HTML5 is optional and there is no need to use it at all since all browsers have always assumed the correct default (even in HTML 4 where it was required by the specification).
- `text/plain`: Textual data; Defined in RFC 2046 and RFC 3676
- `text/rtf`: RTF; Defined by Paul Lindner
- `text/vcard`: vCard (contact information); Defined in RFC 6350
- `text/vnd.abc`: ABC music notation; Registered[14]
- `text/xml`: Extensible Markup Language; Defined in RFC 3023

Type `video` [edit]

For video.

- `video/avi` : Covers most Windows-compatible formats including .avi and .divx[15]
- `video/example`: example in documentation, Defined in RFC 4735
- `video/mpeg`: MPEG-1 video with multiplexed audio; Defined in RFC 2045 and RFC 2046
- `video/mp4`: MP4 video; Defined in RFC 4337
- `video/ogg`: Ogg Theora or other video (with audio); Defined in RFC 5334
- `video/quicktime`: QuickTime video; Registered[16]
- `video/webm`: WebM Matroska-based open media format
- `video/x-matroska`: Matroska open media format
- `video/x-ms-wmv`: Windows Media Video; Documented in Microsoft KB 288102
- `video/x-flv`: Flash video (FLV files)

List of common media subtype prefixes [edit]

Prefix `vnd` [edit]

For vendor-specific files.

- `application/vnd.oasis.opendocument.text`: OpenDocument Text; Registered[17]
- `application/vnd.oasis.opendocument.spreadsheet`: OpenDocument Spreadsheet; Registered[18]
- `application/vnd.oasis.opendocument.presentation`: OpenDocument Presentation; Registered[19]
- `application/vnd.oasis.opendocument.graphics`: OpenDocument Graphics; Registered[20]
- `application/vnd.ms-excel`: Microsoft Excel files
- `application/vnd.openxmlformats-officedocument.spreadsheetml.sheet`: Microsoft Excel 2007 files
- `application/vnd.ms-powerpoint`: Microsoft Powerpoint files
- `application/vnd.openxmlformats-officedocument.presentationml.presentation`: Microsoft Powerpoint 2007 files
- `application/vnd.openxmlformats-officedocument.wordprocessingml.document`: Microsoft Word 2007 files
- `application/vnd.mozilla.xul+xml`: Mozilla XUL files
- `application/vnd.google-earth.kml+xml`: KML files (e.g. for Google Earth)[21]
- `application/vnd.google-earth.kmz`: KMZ files (e.g. for Google Earth)[22]
- `application/dart`: Dart files [23]
- `application/vnd.android.package-archive`: For download apk files.
- `application/vnd.ms-xpsdocument`: XPS document[24]
- `text/vnd.abc`: ABC music notation; Registered[14]

Prefix `x` [edit]

For experimental or non-standard files. Deprecated by RFC 6648 .[25]

- `application/x-7z-compressed`: 7-Zip compression format.
- `application/x-chrome-extension`: Google Chrome/Chrome OS extension, app, or theme package [26]
- `application/x-deb`: deb (file format), a software package format used by the Debian project
- `application/x-dvi`: device-independent document in DVI format
- `application/x-font-ttf`: TrueType Font No registered MIME type, but this is the most commonly used
- `application/x-javascript`:
- `application/x-latex`: LaTeX files
- `application/x-mpegURL`: .m3u8 variant playlist
- `application/x-rar-compressed`: RAR archive files
- `application/x-shockwave-flash`: Adobe Flash files for example with the extension .swf
- `application/x-stuffit`: StuffIt archive files
- `application/x-tar`: Tarball files
- `application/x-www-form-urlencoded`: Form Encoded Data; Documented in HTML 4.01 Specification, Section 17.13.4.1
- `application/x-xpinstall`: Add-ons to Mozilla applications (Firefox, Thunderbird, SeaMonkey, and the discontinued Sunbird)
- `audio/x-aac`: .aac audio files
- `audio/x-caf`: Apple's CAF audio files
- `image/x-xcf`: GIMP image file
- `text/x-gwt-rpc`: GoogleWebToolkit data
- `text/x-jquery-tmpl`: jQuery template data
- `text/x-markdown`: Markdown formatted text
- `application/x-pkcs12`: a variant of PKCS standard files

See also [edit]

- XML and MIME
- `mime.types` file
- Content sniffing

References [edit]

1. ^ a b "Internet Media Type registration, consistency of use" . W3C. 2002-09-04. Retrieved 2012-02-29.
2. ^ "MIME Type Detection in Windows Internet Explorer" . Microsoft. Retrieved 2012-07-14.
3. ^ a b Gordon P. Hemsley, Adam Barth, Ian Hickson (29 November 2012). "MIME Sniffing Standard, Living Standard" . Mimesniff.spec.whatwg.org. Retrieved 2013-08-16.
4. ^ "CVE-2008-5343 (under review)" . MITRE Corporation. 4 December 2008. Retrieved 1 January 2013.

5. ^ Henry Sudhof (11 February 2009). "Risky sniffing: MIME sniffing in Internet Explorer enables cross-site scripting attacks" . The H. Retrieved 2012-07-14.
6. ^ *The 'Codecs' and 'Profiles' Parameters for "Bucket" Media Types* , 2011-08
7. ^ *Structured Syntax Suffix Registry* , 2012-07-20
8. ^ RFC 2046 - Multipurpose Internet Mail Extensions (MIME) Part Two: Media types. Tools.ietf.org. Retrieved on 2010-09-29.
9. ^ W3C (June 1999). "RFC 2616: 7. Entity" . *Hypertext Transfer Protocol -- HTTP/1.1*. The Internet Society. Retrieved 28 May 2012.
10. ^ MIME SUBTYPE NAME: zip
11. ^ "Supported Media Formats" . *RealPlayer Help*. RealNetworks. 2010. Retrieved 28 May 2012.
12. ^ MIME SUBTYPE NAME: png
13. ^ http://www.djvu.org
14. ^ *a* *b* vnd.abc
15. ^ Mime Types List
16. ^ Quicktime
17. ^ vnd.oasis.opendocument.text
18. ^ vnd.oasis.opendocument.spreadsheet
19. ^ vnd.oasis.opendocument.presentation
20. ^ vnd.oasis.opendocument.graphics
21. ^ "Application Media Types" . IANA. Retrieved 2012-02-19.
22. ^ "KML Tutorial - Keyhole Markup Language — Google Developers". Developers.google.com. Retrieved 2013-08-16.
23. ^ "Embedding Dart in HTML". Dartlang.org. Retrieved 2013-12-03.
24. ^ "MIME Type" . Social.msdn.microsoft.com. Retrieved 2013-08-16.
25. ^ "RFC 6648 - Deprecating the "X-" Prefix and Similar Constructs in Application Protocols" . IETF. June 2012. Retrieved 2012-10-07.
26. ^ http://developer.chrome.com/extensions/hosting.html

External links [edit]

- IANA MIME media types list
- IANA character sets
- RFC 2045 , RFC 2046 - Multipurpose Internet Mail Extensions (MIME), parts 1 and 2
- RFC 4288 - Media Type Specifications and Registration Procedures

Categories: Computer file formats | Metadata

This page was last modified on 2 May 2014 at 09:41.

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.

Privacy policy About Wikipedia Disclaimers Contact Wikipedia Developers Mobile view